Namal University, Mianwali

Department of Electrical Engineering

Communication Systems (Lab)

Lab – 3

Amplitude Modulation and Demodulation using MATLAB/Simulink

Student Name	Student ID
Fahim Ur Rehman Shah	NIM-BSEE-2021-24

Instructor: Dr. Sajjad Ur Rehman

Lab Engineer: Faizan Ahmad

Version 1.0 – 27/03/2021 – Engr. Arbaz
Version 1.1 – 27/03/2022 – Engr. Arbaz
Version 1.2 – 20/03/2023 – Engr. Imtiaz Ul Hassan
Version 1.3 – 26/02/2024 – Engr. Faizan Ahmad

Introduction

The purpose of this lab is to provide the students basic understanding of DSB-SC Amplitude Modulation and Demodulation in MATLAB/Simulink.

Course Learning Outcomes

CLO2: Develop software simulations to observe the performance of analog and digital communication systems.
CLO4: Report desired results proofs and calculations.

Equipment

- Software
  - MATLAB

Instructions

- This is an individual lab. You will perform the tasks individually and submit the required files at the end of the lab.
- Plagiarism or any hint thereof will be dealt with strictly. Any incident where plagiarism is caught, both (or all) students involved will be given zero marks, regardless of who copied whom. Multiple such incidents will result in disciplinary action being taken.

Amplitude Modulation:

Amplitude Modulation (AM) is a technique used in electronic communication, most commonly for transmitting information via a carrier signal. AM works by varying the strength of the transmitted signal in relation to the information being sent. For example, changes in the modulated signal strength can be used to obtain sounds to be reproduced by a loudspeaker, or the light intensity of television pixels.

An amplitude modulated signal can be described as follows

$$y_{AM}(t) = m(t) * \cos(2\pi f_c t)$$

Here m (t) is the message signal to be transmitted, f_c is the carrier frequency and y_{AM} (t) is the amplitude modulated signal. Let the message be a time-varying signal such as

$$m(t) = A_m \cos(2\pi f_m t)$$

For demodulation we multiply modulated signal with carrier and then use low pass filter to extract message signal.

$$y_{DE\text{-}Mod}(t) = y_{AM} * \cos(2\pi f_c t)$$

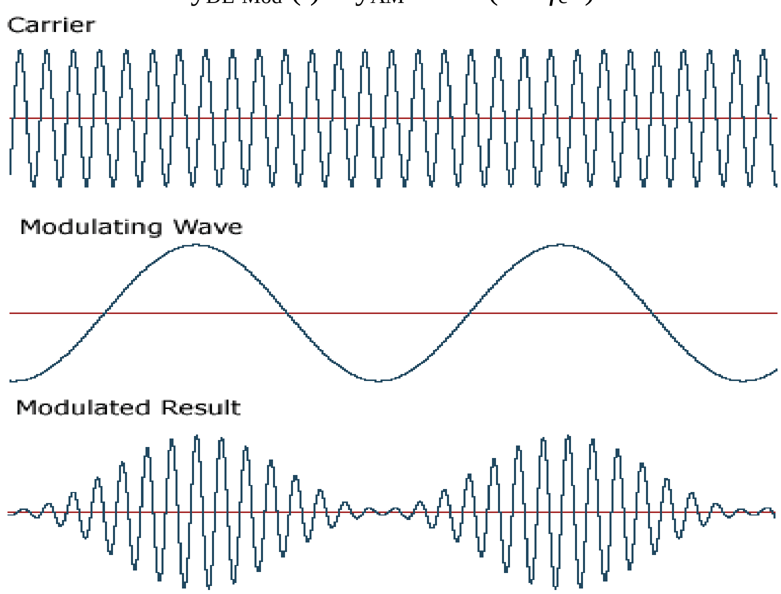


Figure 1: Carrier, Modulating and Modulated Signals

EXERCISE:

Task 01

Write MATLAB Code for DSB Modulation

$$y_{DSB} = A_m \cos(2\pi f_m t)\, V_c \cos(2\pi f_c t)$$

Where $A_m = 1$ and $f_m = 100Hz$, $V_c = 2$ and $f_c = 10kHz$

- Plot $m(t)$, $V_c \cos(2\pi f_c t)$ and y_{DSB} in time domain for $t = 0 : 1/\text{fs}:1$, where $f_s = 1000Hz$
- Plot frequency spectrum of m(t), $V_c cos(2\pi f_c t)$ and y_{DSB} in frequency domain and reconstruct demodulated signal using filter.
- What can you analyze from the frequency components? Are they consistent with your theoretical knowledge?

Task 02

Implement DSB-SC Modulation and De-modulation using Simulink, you can use the following blocks in Simulink to implement it. Where $f_{\mathrm{m}} = 10Hz$ and $f_{\mathrm{c}} = 100Hz$

- Signal Generator (Carrier and Message Generation)
- Dot Product (for multiplication)
- Analog Filter Design (Butter-low pass)
- Gain (use at Demodulation side)
- Scope

Report

For each exercise, include the code you write, explanation where asked, all outputs, as well as answers to any questions, as required. Upload your final report (containing general formalities like cover page containing your name and roll number etc.) to QOBE in the report submission folder.

Upload your final executable .m file as well in the separate submission folder on QOBE.

EXERCISE:

Task 01

Write MATLAB Code for DSB Modulation

$$yDSB = Am\cos(2\ fmt)\ Vc\cos(2\ fct)$$

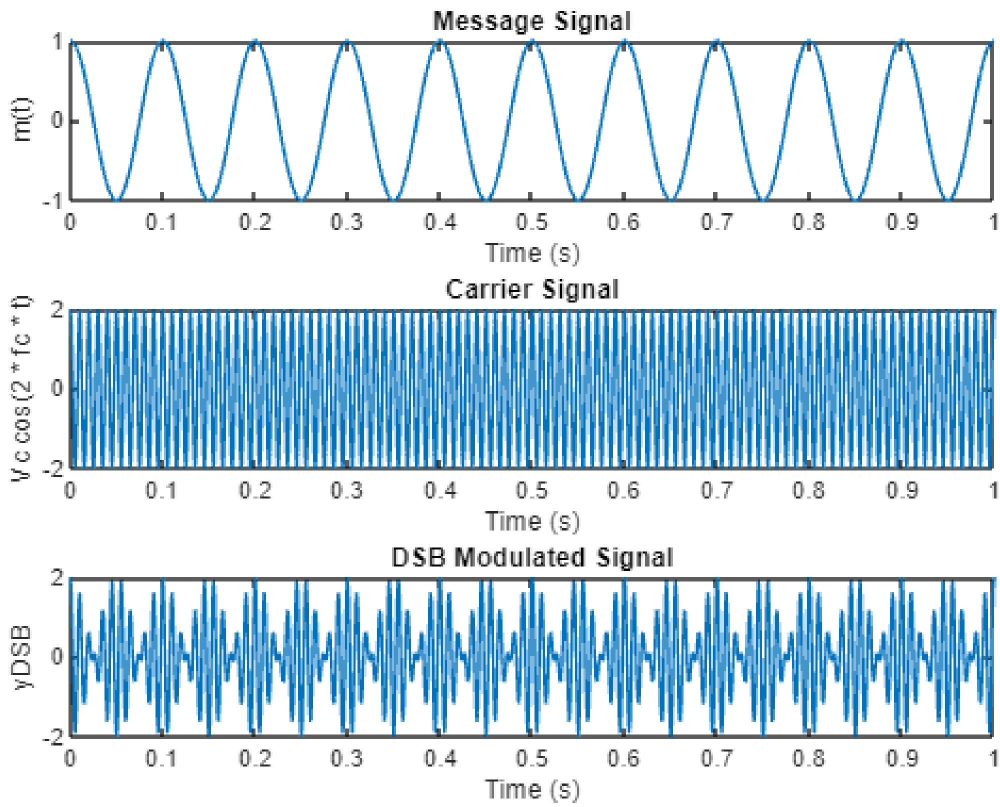
Where $Am = 1$ and $fm = 100Hz$, $Vc = 2$ and $fc = 10kHz$

- Plot $m(t)$, Vc cos (2 fc t) and yDSB in time domain for $t = 0 : 1/$fs:1, where $fs = 1000Hz$
- Plot frequency spectrum of m(t), $Vccos(2\ fct)$ and $yDSB$ in frequency domain and reconstruct demodulated signal using filter.
- What can you analyze from the frequency components? Are they consistent with your theoretical knowledge?

```
% Parameters
Am = 1;
fm = 10; % in Hz
Vc = 2;
fc = 100; % in Hz
fs = 1000; % in Hz
t = 0:1/fs:1;

% Signals
mt = Am * cos(2 * pi * fm * t);
Vc_cos = Vc * cos(2 * pi * fc * t);
yDSB = Am * cos(2 * pi * fm * t) .* Vc_cos;

% Plot time domain signals
figure;
subplot(3,1,1);
plot(t, mt);
xlabel('Time (s)');
ylabel('m(t)');
title('Message Signal');
subplot(3,1,2);
plot(t, Vc_cos);
xlabel('Time (s)');
ylabel('Vc cos(2 * fc * t)');
%xlim([0 0.2])
title('Carrier Signal');
subplot(3,1,3);
plot(t, yDSB);
xlabel('Time (s)');
ylabel('yDSB');
title('DSB Modulated Signal');
```

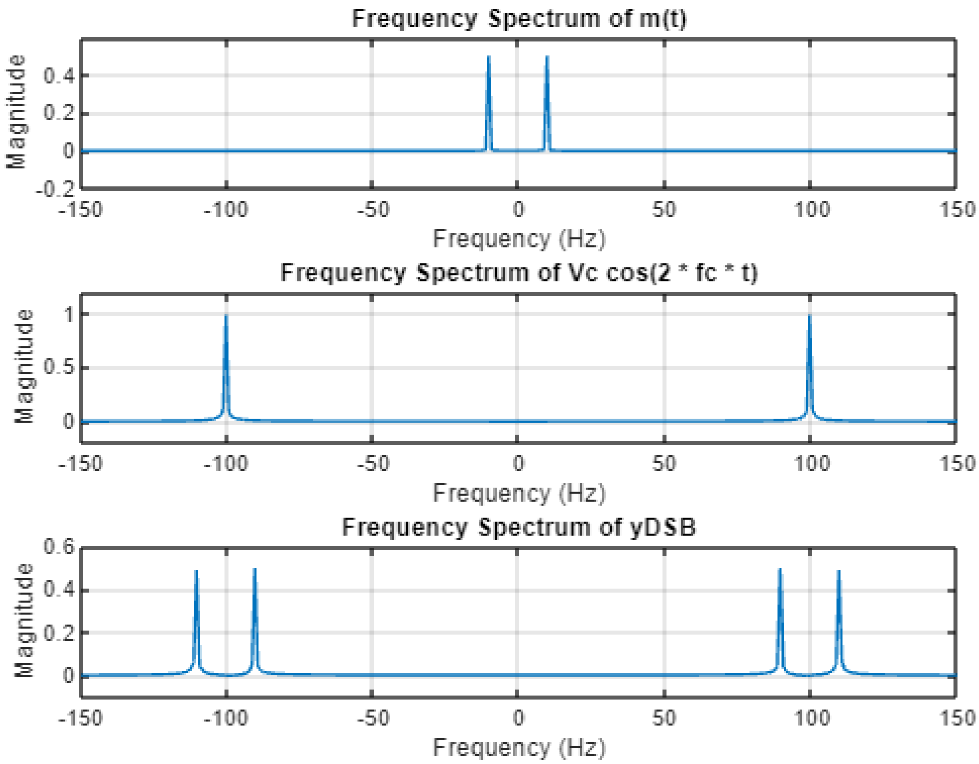



```
%xlim([0 0.5])

% Plot frequency domain signals
f = -fs/2:fs/length(t):fs/2-fs/length(t);
MT = fftshift(fft(mt));
VC_COS = fftshift(fft(Vc_cos));
YDSB = fftshift(fft(yDSB));
figure;
subplot(3,1,1);
plot(f, abs(MT)/length(t));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Spectrum of m(t)');
xlim([-150 150])
grid("on")
ylim([-0.2 0.6])

subplot(3,1,2);
plot(f, abs(VC_COS)/length(t));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Spectrum of Vc cos(2 * fc * t)');
xlim([-150 150])
ylim([-0.2 1.2])
grid("on")
```

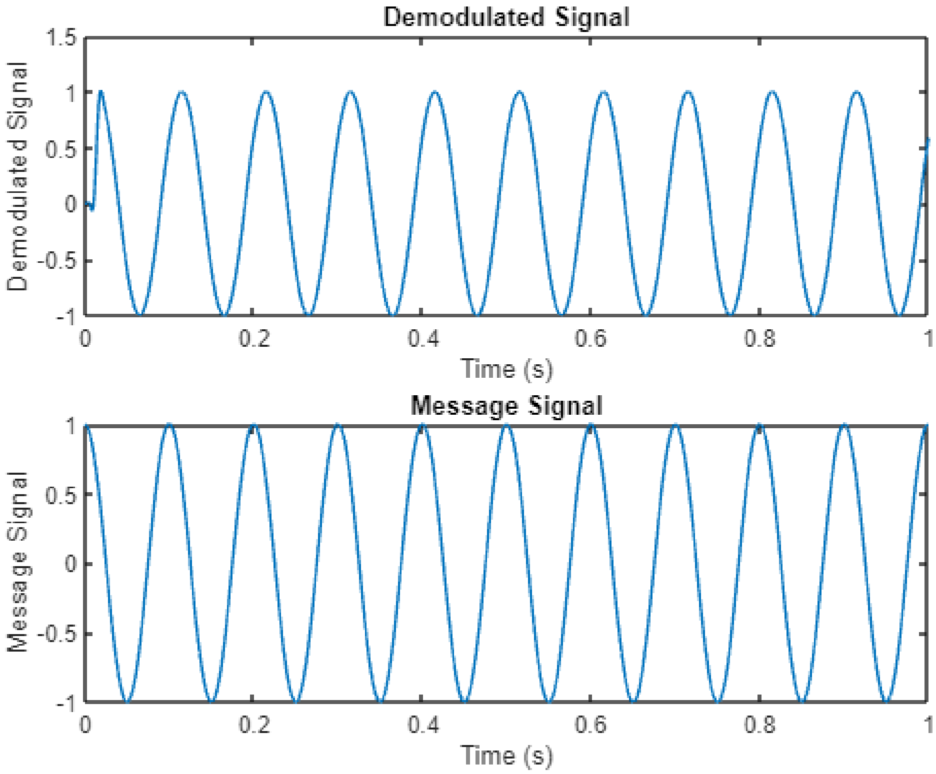
```
subplot(3,1,3);
plot(f, abs(YDSB)/length(t));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Spectrum of yDSB');
xlim([-150 150])
ylim([-0.1 0.6])
grid("on")
```




```
% Reconstruct demodulated signal using filter
y1=yDSB.*Vc_cos;
h = fir1(30,   fc/(fs/2)); % Design a lowpass filter
y_demod =0.5* filter(h, 1, y1); % Demodulate using the filter

% Plot demodulated signal
figure;
subplot(2,1,1)
plot(t, y_demod);
xlabel('Time (s)');
ylabel('Demodulated Signal');
title('Demodulated Signal');
subplot(2,1,2)
plot(t, mt);
xlabel('Time (s)');
```

```
ylabel('Message Signal');
title('Message Signal');
```



Explanation:

I set up some parameters for a communication system. I have a message signal, a carrier signal, and a modulated signal. Then, I plotted these signals to see how they look over time and in terms of their frequency. After that, I demodulated the modulated signal using a filter to recover the original message signal. Finally, I compared the demodulated signal with the original message signal by plotting them together. This code helps me understand how communication signals are processed and how modulation and demodulation work in practical scenarios.

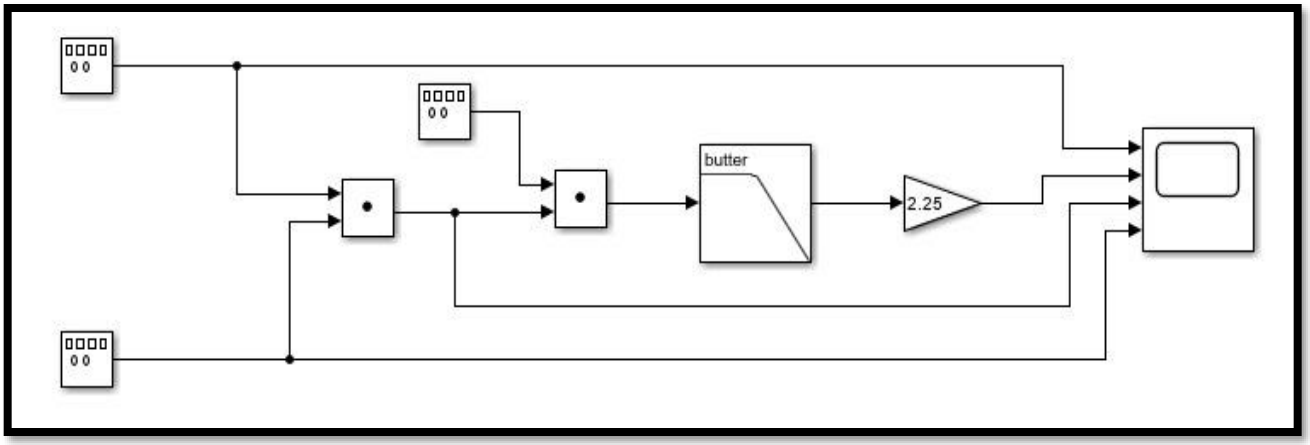
Task 02

Implement DSB-SC Modulation and De-modulation using Simulink, you can use the following blocks in Simulink to implement it.

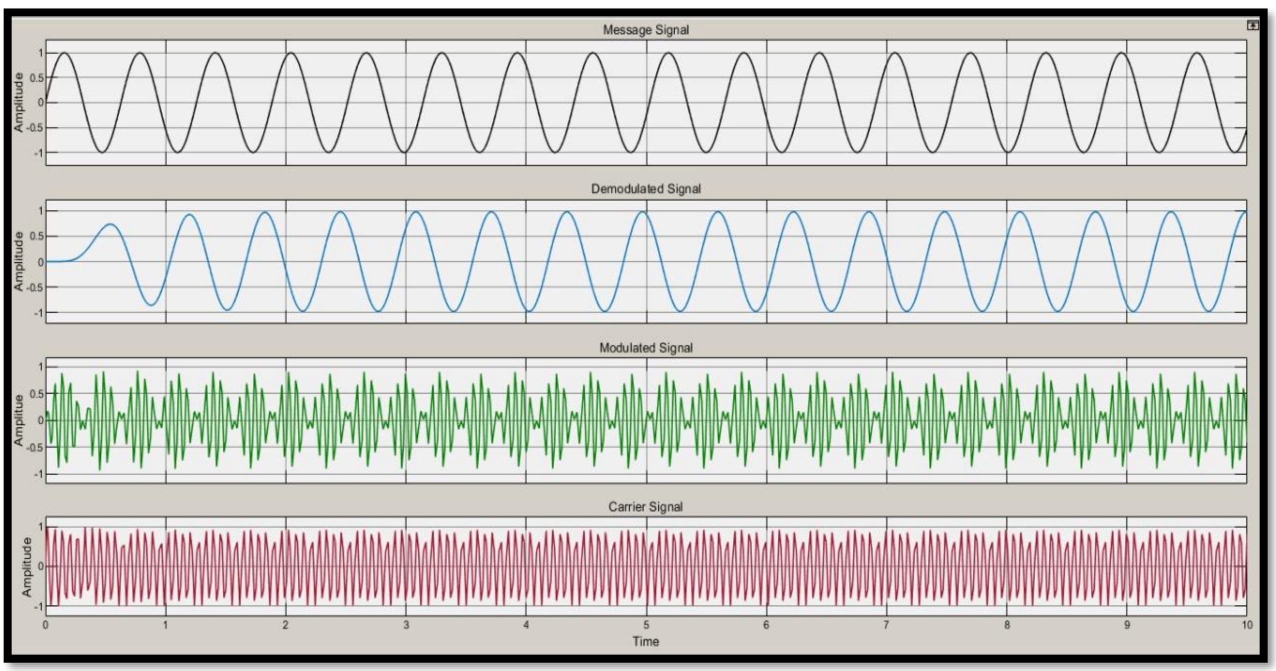
Where fm = 10Hz and fc = 100Hz

- Signal Generator (Carrier and Message Generation)
- Dot Product (for multiplication)
- Analog Filter Design (Butter-low pass)
- Gain (use at Demodulation side)
- Scope

Block Diagram:



Output:



Explanation:

To implement DSB-SC (Double Sideband Suppressed Carrier) Modulation and De-modulation using Simulink, I used several blocks: Signal Generator for generating both carrier and message signals with frequencies of 10 Hz and 100 Hz respectively. Then, I used the Dot Product block for multiplying the message signal with the carrier signal. Next, I utilized the Analog Filter Design block, specifically a Butterworth low-pass filter, to filter the modulated signal. At the demodulation side, I used the Gain block to adjust the amplitude of the demodulated signal. Finally, I used the Scope block to visualize the signals at various stages of modulation and demodulation. This setup in Simulink demonstrated the process of DSB-SC modulation and demodulation.

Conclusion:

In conclusion, I successfully implemented DSB-SC (Double Sideband Suppressed Carrier) Modulation and De-modulation using Simulink. I utilized various blocks within Simulink, including Signal Generator, Dot Product, Analog Filter Design, Gain, and Scope, to carry out the modulation and demodulation processes. By generating carrier and message signals, multiplying them, filtering the modulated signal, and adjusting the amplitude of the demodulated signal, I effectively demonstrated the principles of DSB-SC modulation and demodulation. This exercise enhanced my understanding of communication systems and signal processing techniques, providing valuable insights into practical applications of modulation and demodulation in engineering and telecommunications.

Com. Sys. Lab 3 Rubric

Method of Evaluation: Executable code, Report submitted by students **Measured Learning Outcomes**:

CLO2: Develop software simulations to observe the performance of analog and digital communications systems.

CLO4: Report desired results proofs and calculations.

	Excellent 10	Good 9-7	Satisfactory 6-4	Unsatisfactory 3-1	Poor 0	Marks Obtained
Code (CLO2)	Correct code, easily understandable with comments where necessary	Correct code but without proper indentation or comments	Slightly incorrect code with proper comments	Incorrect code with improper format and no comments	Code not submitted	
Output (CLO2)	Output correctly shown with all Figures/ Plots displayed as required and properly labelled	Most Output/ Figures/ Plots displayed with proper labels	Some Output/ Figures/ Plots displayed with proper labels OR Most Output/ Figures/ Plots displayed but without proper labels	Most of the required Output/ Figures/ Plots not displayed	Output/ Figures/ Plots not displayed	
Answers (CLO2)	Meaningful answers to all questions. Answers show the understanding of the student.	Meaningful answers to most questions.	Some correct/ meaningful answers with some irrelevant ones	Answers not understandable/ not relevant to questions	Wrong Answers	
Lab Report (CLO4)	Report submitted with proper grammar and punctuation with proper conclusions drawn and good formatting	Report submitted with proper conclusions drawn with good formatting but some grammar mistakes OR proper grammar but not very good formatting	Some correct/ meaningful conclusions. Some parts of the document not properly formatted or some grammar mistakes	Conclusions not based on results. Bad formatting with no proper grammar/ punctuation	Report not submitted	
Total						